

Amendments to the Claims:

1. (Original) A method for assessing a functional size of a software application or project including the steps of:

analysing a software requirements specification and determining zero or more keywords for each requirement of the specification;

using a computer to cross-reference the keywords with a lexicon stored in a computer file, said lexicon also including a function type and complexity for each keyword, and further using the computer to associate each keyword with an entry in the lexicon, thus obtaining a function type and complexity for each keyword;

using a functional sizing standard to deduce a number of function points for the function type and complexity of each keyword; and

combining the function points to obtain a functional size of the software application or project.

2. (Original) The method of claim 1 wherein:

the step of analysing the requirements specification includes parsing each requirement to at least isolate lexical elements and determining at least one keyword corresponding to each lexical element; and

the step of cross-referencing a keyword with the lexicon includes searching the lexicon for an entry which matches said keyword.

3. (Original) The method of claim 2 wherein said lexical elements are individual words of a requirement.

4. (Currently Amended) The method of ~~any one of the preceding claims~~claim 1 wherein determining keywords for a requirement includes identifying words in the requirement that appear in the lexicon.

5. (Currently Amended) The method of ~~any one of the preceding claims~~claim 1 wherein the sizing standard is a standard maintained by the International Function Point Users Group (IFPUG).

6. (Currently Amended) The method of ~~any one of the preceding claims~~claim 1 wherein:

the function type associated with a keyword in the lexicon is one of Internal Logical File, External Interface File, External Input, External Output or External Enquiry; and

the complexity associated with each keyword is one of Low Complexity, Average Complexity or High Complexity.

7. (Currently Amended) The method of ~~any one of the preceding claims~~claim 1 wherein the step of combining the function points includes summing the function points associated with all identified keywords to obtain a functional size that is equal to the total number of function points.

8. (Currently Amended) The method of ~~any one of the preceding claims~~claim 1, further including the step of deducing from the functional size at least one parameter associated with management of the development of the software application.

9. (Original) The method of claim 8 wherein said at least one parameter is one or more of cost, effort, and/or development time.

10. (Currently Amended) The method of claim 8 or 9 wherein the step of deducing the at least one parameter includes multiplying the functional size by a number representing a corresponding productivity rate.

11. (Original) A computer implemented system for assessing a functional size of a software application or project, wherein the system receives a software requirements specification as input and includes:

a lexicon stored in a computer file, the lexicon including keywords and a function type

and complexity for each keyword;

computer instruction code for analysing the software requirements specification to associate each requirement with zero or more requirement keywords;

computer instruction code for cross-referencing the requirement keywords with the lexicon to obtain a corresponding function type and complexity from the lexicon by matching requirement keywords with lexicon keywords;

computer instruction code for computing a number of function points associated with each function type and complexity using the rules of a sizing standard; and

computer instruction code for combining the function points to obtain a functional size of the software application or project.

12. (Original) The system of claim 11 wherein the software requirements specification is received as input by a user performing a cut and paste operation from a source application.

13. (Original) The system of claim 11 wherein the software requirements specification is received as input from a file stored on a non-volatile storage medium of a personal computer.

14. (Original) The system of claim 11 wherein the system is connected to a communications network and the software requirements specification is received as input from a remote source over the network.

15. (Currently Amended) The system of ~~any one of claims 11 to 14~~claim 11 further including a lexicon editor for enabling a user to modify the computer file in which the lexicon is stored, such that the lexicon keywords and corresponding function type and complexity can be changed, and new lexicon keywords and their corresponding function type and complexity can be added.

16. (Currently Amended) The system of ~~any one of claims 11 to 15~~claim 11 wherein the lexicon keywords are words that may appear in requirements of the requirements specification.

17. (Currently Amended) The system of ~~any one of claims 11 to 16~~claim 11 wherein the function type associated with a keyword and stored in the lexicon is one of Internal Logical File, External Interface File, External Input, External Output or External Enquiry and the complexity associated with each keyword and stored in the lexicon is one of Low Complexity, Average Complexity or High Complexity.

18. (Original) In a networked computing system including a client and a server both of which are operably connected to a communications network, a method for assessing a functional size of a software application or project including the steps of:

the client transmitting a software requirements specification to the server over the communications network; and

the client receiving a functional size of the software application or project from the server over the communications network,

wherein the server executes the steps of:

analysing a software requirements specification and determining zero or more keywords for each requirement of the specification;

cross-referencing the keywords with a lexicon stored in a computer file, said lexicon also including a function type and complexity for each keyword;

associating each keyword with an entry in the lexicon, thus obtaining a function type and complexity for each keyword;

using a functional sizing standard to deduce a number of function points for the function type and complexity of each keyword; and

combining the function points to obtain a functional size of the software application or project.

19. (Original) A computer program product for assessing a functional size of a software application or project including computer instruction code embodied in a computer readable medium for:

analysing a software requirements specification and determining zero or more keywords for each requirement of the specification;

cross-referencing the keywords with a lexicon stored in a computer file, said lexicon also including a function type and complexity for each keyword;

associating each keyword with an entry in the lexicon, thus obtaining a function type and complexity for each keyword;

using a functional sizing standard to deduce a number of function points for the function type and complexity of each keyword; and

combining the function points to obtain a functional size of the software application or project.

20. (Original) The computer program product of claim 19 wherein:

analysing the requirements specification includes parsing each requirement to isolate lexical elements and determining at least one keyword corresponding to each lexical element; and

cross-referencing a keyword with the lexicon includes searching the lexicon for an entry which matches said keyword.

21. (Original) The computer program product of claim 20 wherein said lexical elements are individual words of a requirement.

22. (Currently Amended) The computer program product of claim 20 or ~~claim 21~~ wherein determining keywords for a requirement includes identifying words in the requirement that appear in the lexicon.

23. (Currently Amended) The computer program product of ~~any one of claims 19 to 22~~ claim 19 wherein combining the function points includes summing the function points

associated with all identified keywords to obtain a functional size that is equal to the total number of function points.

24. (Currently Amended) The computer program product of ~~any one of claims 19 to 23~~claim 19, further including computer instruction code embodied on the computer readable medium for deducing from the functional size at least one parameter associated with management of the development of the software application.

25. (Original) The computer program product of claim 24 wherein said at least one parameter is one or more of cost, effort, and/or development time.

26. (Currently Amended) The computer program product of claim 24 or 25 wherein deducing the at least one parameter includes multiplying the functional size by a number representing a corresponding productivity rate.

27. (Currently Amended) The computer program product of ~~any one of claims 19 to 26~~claim 19 wherein the software requirements specification is received as input using a virtual clipboard of a computer executing the computer instruction code by a user performing a cut and paste operation from a source application.

28. (Currently Amended) The computer program product of ~~any one of claims 19 to 27~~claim 19 further including computer instruction code embodied on the computer readable medium for providing a lexicon editor that enables a user to modify the computer file in which the lexicon is stored, such that the lexicon keywords and corresponding function type and complexity can be changed, and new lexicon keywords and their corresponding function type and complexity can be added.